DOCUMENT RESUME

ED 303 137 IR 013 591

AUTHOR Gelman, Michael A.
TITLE Crazy Critters: Creating New Life Forms in Logo.
PUB DATE 7 May 88
NOTE 17p.; Paper presented at the Great Lakes/East Coast LOGO Conference (Cleveland, OH, May 7, 1988).
PUB TYPE Computer Programs (101) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Computer Games; Computer Graphics; *Computer Software; Junior High Schools; *Microcomputers; Programing; Programing Languages
IDENTIFIERS *LOGO Programing Language

ABSTRACT

This paper lists the commands for three programs in Logo: Peppy and the Maze, Caterpillars and Butterflies, and the Robot Arm. (EW)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED303137

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

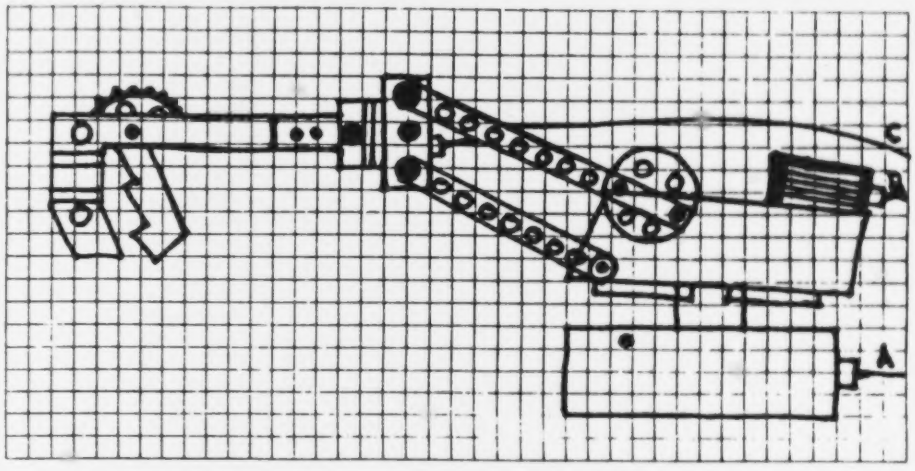
☒ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

CRAZY CRITTERS:

Creating New Life Forms in Logo



Michael A. Gelman

Schuyler Colfax Jr. High School

Wayne, New Jersey

Great Lakes/East Coast LOGO Conference

Cleveland, Ohio

May 7, 1988

(c) 1988 Michael A. Gelman

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Michael A. Gelman

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

IR013591

BEST COPY AVAILABLE

CRAZY CRITTERS:

Creating New Life Forms in Logo

1. Peppy and the Maze
2. Caterpillars and Butterflies
3. The Robot Arm

Michael A. Gelman
19 Darlington Drive
Wayne, NJ 07470-2805
(201) 595-0319

```
to startup
getshapes
gettools "peppysys
end

to maze
cg
pu setc 3
setpos [-125 -50]
seth 0
block 8
rt 90
block 15
rt 90
block 8
rt 90
block 4
rt 90
block 5
lt 90
block 6
lt 90
block 4
rt 90
block 2
pu
fd 48
```



```
rt 90
fd 32
rt 90
pd
block 2
end

to robot
tell 1
pu
st
setc 2
setpos [85 -10]
seth (random 4) * 90
shape.change
end

to food
pu setsh 7
setc 1
setpos [-107 -60]
pd
stamp
pu
setpos [-88 -60]
pd
stamp
```



```
end

to threshold
op 1523
end
```

PEPPY 2 PROGRAM 2.1

```
to startup
getshapes
gettools "peppysys
end

to maze
cg
pu setc 3
setpos [-125 -50]
seth 0
block 8
rt 90
block 15
rt 90
block 8
rt 90
block 4
rt 90
block 5
lt 90
block 6
lt 90
block 4
rt 90
block 2
pu
fd 48
```

PEPPY 2 PROGRAM 2.2

```
rt 90
fd 32
rt 90
pd
block 2
pu
seth 0
fd 32
rt 90
fd 32
rt 180
block 1
end

to robot
tell 1
pu
st
setc 2
setpos [85 -10]
seth (random 4) * 90
shape.change
end

to food
pu setsh 7
setc 1
```

PEPPY 2 PROGRAM 2.3

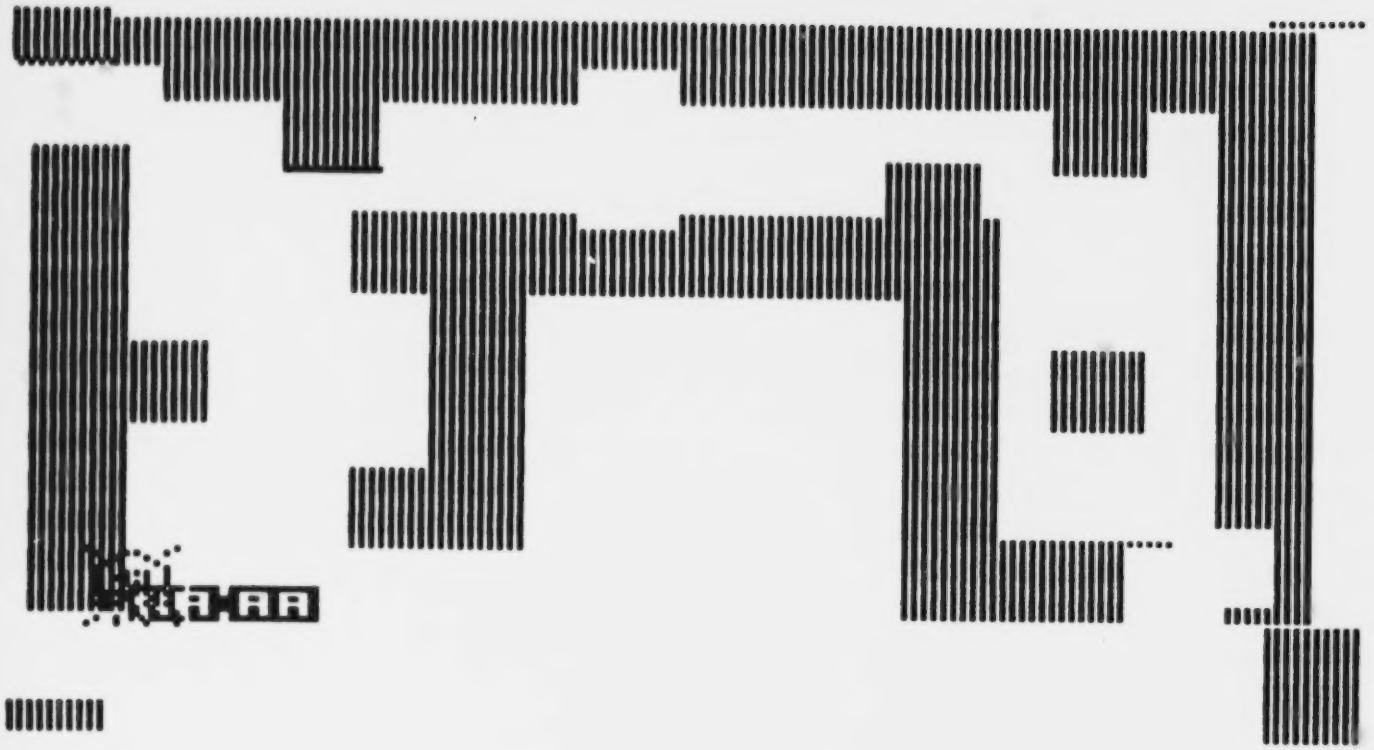
```
setpos [-107 -60]
pd
stamp
pu
setpos [-88 -60]
pd
stamp
end

to threshold
op 1785.6
end
```

```
to block :number
setsh 11
pd
repeat :number [stamp pu fd 16 pd]
end

to shape.change
setsh (heading / 90) + 3
end

to move :moves.used
if colorunder = 1 [type (se [You used]
    :moves.used [moves.]) show " win
    stop]
if colorunder = 3 [bk 8 lt (1 + random
    3) * 90 shape.change]
fd 4
if :moves.used > threshold [slowpoke
    stop]
move :moves.used + 1
end

to peppy
rg
maze
food
robot
```



```
move 0
end

to win
setsh 8
tell 3 ht pu setpos [-130 -75]
repeat 20 [label [Congratulations!]
    wait 5]
end

to startup
rg
ct
end

to robot
tell 1
pu
st
setc 2
setpos [85 -10]
seth (random 4) * 90
shape.change
end

to food
pu setsh 7
```

```
setc 1
setpos [-107 -60]
pd
stamp
pu
setpos [-88 -60]
pd
stamp
end

to threshold
op 1523
end

to mazetest
food
robot
move 0
end

to slowpoke
setsh 9
type [Slowpoke! Your batteries ran
     out.]
end
```



PEPPY PRISON 4.1

```
to block :number
setsh 11
pd
repeat :number [stamp pu fd 16 pd]
end

to shape.change
setsh (heading / 90) + 3
end

to move :moves.used
tell 1
if colorunder = 1 [type (se [You used]
    :moves.used [moves.]) show " win
    stop]
if colorunder = 3 [bk 8 lt (1 + random
    3) * 90 shape.change]
fd 4
if hittingp 2 [eat]
move.predator
if :moves.used > threshold [slowpoke
    stop]
move :moves.used + 1
end

to peppy
rg
```



```
maze
food
robot
pred.set
move 0
end

to win
setsh 8
tell 3 ht pu setpos [-130 -75]
repeat 20 [label [Congratulations!]
    wait 5]
tell 2 setsh 17
end

to eat
tell 2
setsh 16
tell 3 ht pu setpos [-130 -75]
repeat 20 [label [Fresh AA batteries!
  My favorite!!!] wait 5]
tell 1 ht
tell 2 setsh 17
end

to startup
rg
```

```
ct
end

to robot
tell 1
pu
st
setc 2
setpos [85 -10]
seth (random 4) * 90
shape.change
end

to food
pu setsh 7
setc 1
setpos [-107 -60]
pd
stamp
pu
setpos [-88 -60]
pd
stamp
end

to threshold
op 1523
```



```
end

to mazetest
food
robot
pred.set
move 0
end

to slowpoke
setsh 9
type [Slowpoke! Your batteries ran
     out.]
end

to pred.set
tell 2
setsh 12
setc 3
pu st
setpos [-80 30]
end

to turtle.pos :who
op ask :who [pos]
end
```

```
to hittingp :who
op 17 > distance pos turtle.pos :who
end

to move.predator
tell 2
make "peppy.pos turtle.pos 1
make "pred.pos turtle.pos 2
seth (90 * (round ((towards
   :peppy.pos) / 90)))
pred.change
fd 5
if colorunder = 3 [bk 10 stop]
if hittingp 1 [eat]
end

to pred.change
setsh (12 + (heading / 90))
end

to distance :pt1 :pt2
make "pt1x first :pt1
make "pt2x first :pt2
make "pt1y last :pt1
make "pt2y last :pt2
make "difx :pt1x - :pt2x
make "dify :pt1y - :pt2y
```



```
op sqrt ((:difx * :difx) + (:dify *
  :dify))
end
```

```
to startup
getshapes
gettools "peppy.predsys
end

to maze
cg
pu setc 3
setpos [-125 -50]
seth 0
block 8
rt 90
block 15
rt 90
block 8
rt 90
block 4
rt 90
block 5
lt 90
block 6
lt 90
block 4
rt 90
block 2
pu
fd 48
```



```
rt 90
fd 32
rt 90
pd
block 2
pu
seth 0
fd 32
rt 90
fd 32
rt 180
block 1
end

to robot
tell 1
pu
st
setc 2
setpos [85 -10]
seth (random 4) * 90
shape.change
end

to food
pu setsh 7
setc 1
```



```
setpos [-107 -60]
pd
stamp
pu
setpos [-88 -60]
pd
stamp
end

to threshold
op 1785.6
end
```

```
to startup
gettools "peppy.pledgesys
end

to maze
cg
pu setc 3
setpos [-125 -50]
seth 0
block 3
rt 90
block 3
lt 90
block 3
pu
bk 48
rt 90
bk 48
lt 90
pd
block 5
rt 90
block 8
rt 90
block 3
rt 90
pu
```



```
fd 48
rt 180
pd
block 6
pu
bk 48
rt 90
bk 48
lt 90
pd
block 7
rt 90
block 4
rt 90
block 4
pu
bk 64
lt 90
pd
block 4
rt 90
block 7
rt 90
pu
fd 32
rt 90
block 6
```



```
pu
bk 96
lt 90
bk 32
lt 90
block 7
end

to robot
tell 1
pu
st
setc 2
setpos [85 -40]
seth (random 4) * 90
shape.change
end
```

```
to block :number
setsh 11
pd
repeat :number[stamp pu fd 16 pd]
end

to food
pu setsh 7
setc 1
setpos [-110 -60] pd stamp
pu setpos [-91 -60]
pd stamp
end

to robot
tell 1 st
pu setpos [-15 0]
setsh 3 setc 2
seth (random 4) * 90
shape.change
end

to shape.change
setsh (heading / 90)+3
end

to free.run :counter
```



```
fd 4
if colorunder = 3 [wall.follow
     (:counter + 1) 0]
if colorunder = 1 [win stop]
free.run :counter + 1
end

to peppy :north
cg maze food robot
shape.change
make "turning 0
make "throw.mark "
ct
free.run 0
end

to win
setsh 8
tell 3 ht pu setpos [-130 -75]
repeat 20[label [Congratulations!] wait
     5 ]
end

to startup
rg ct
end
```

```
to wall.follow :counter :turning
bk 4 lt 90 shape.change
make "turning :turning - 90
wall.run :counter + 1 :turning
if :throw.mark = "north [stop]
end

to wall.run :counter :turning
if :turning = 0 [make "throw.mark
    "north stop]
if colorunder = 1 [win stop]
if not (wall.on.rightp) [rt 90 make
     "turning :turning + 90
     shape.change]
fd 4
ifelse colorunder = 3 [move.predator
     wall.follow :counter + 1 :turning]
     [move.predator wall.run :counter +
     1 :turning]
end

to wall.on.rightp
rt 90 fd 4
make "wallcheck colorunder
bk 4 lt 90
ifelse :wallcheck = 3 [op "true] [op
     "false]
```



```
end

to mazetest :north
food robot
shape.change
make "turning 0
make "throw.mark"
ct
free.run 0
end
```

PHANTOM FISHTANK

CATERPILLARS AND BUTTERFLIES

This obscure set of rules must be seen to be believed. Two kinds of creatures exist in this world - caterpillars and butterflies. Breeding grounds where butterflies can run wild add to the unpredictability. "Caterpillars and Butterflies" creates interesting residual patterns. It will be enjoyed by all ages.

```
TO ARM
MAKE "RECORD "
ARM1
END

TO ARM1
COMMANDS READCHAR
ARM1
end

TO COMMANDS :KEY
IF :KEY = "R [TURN.RIGHT R]
IF :KEY = "L [TURN.LEFT L]
IF :KEY = "U [UP U]
IF :KEY = "D [DOWN D]
IF :KEY = "C [CLOSE C]
IF :KEY = "O [OPEN O]
IF :KEY = "P [AO ]
IF :KEY = "S [AO STOPALL]
END

TO TURN.RIGHT
TTO "A
SETEVEN
ONFOR 10
END

TO TURN.LEFT
TTO "A
SETODD
ONFOR 10
END

TO UP
ao
TTO "B
SETEVEN
ONFOR 1
END

TO DOWN
ao
TTO "B
SETODD
ONFOR 1
END
```



```
TO OPEN
TTO "C
SETEVEN
ONFOR 3
END

TO CLOSE
TTO "C
SETODD
ONFOR 3
END

TO R
MAKE "RECORD WORD :RECORD "R
END

TO L
MAKE "RECORD WORD :RECORD "L
END

TO U
MAKE "RECORD WORD :RECORD "U
END

TO D
MAKE "RECORD WORD :RECORD "D
END

TO O
MAKE "RECORD WORD :RECORD "O
END

TO C
MAKE "RECORD WORD :RECORD "C
END
```